LOCST: A Built-In Self-Test Technique

With its low hardware cost, simple implementation and excellent coverage, this technique promises to meet the needs of a variety of VLSI environments.

Johnny J. LeBlanc, IBM Federal Systems Division

The advent of very large scale integration technologies has increased interest in built-in self-test as a technique for achieving effective and economical testing of VLSI components. As used in this article, the term "built-in self-test" refers to the capability of a device to generate its own test pattern set and to compress the test results into a compact pass-fail indication. Many buit-in self-test techniques have been proposed over the past 10 years, ranging from self-oscillation to functional pattern testing of microprogrammed devices to random-pattern testing (for examples, see papers by Mucha et al.,[1] Sedmak,[2] and McCluskey et al.[3]). These various techniques provide different capabilities for defect detection and self-test execution time. They also impose different requirements for implementation and control.

Benefits to be gained from self-test, however, are common to all implementation techniques and include

- reduced test pattern storage requirements,
- reduced test time, and
- defect isolation to the chip level.

Since test patterns are generated automatically, only self-test initialization, control, and pass-fail comparison patterns need be stored, significantly reducing pattern storage requirements. Test time is reduced because one can use simple hardware devices (e.g., counters or linear-feedback shift registers) to control test execution, rather than retrieving test patterns from storage devices (e.g., disks) and applying them to the component under test. When components with built-in self-test are mounted on higher-level packages, the self-test pass-fail indication provides defect isolation to the chip level (e.g., during card repair testing).

At the IBM Federal Systems Division we have implemented a VLSI built-in self-test technique, which can be incorporated at very low hardware cost into any chip conforming to level-sensitive scan design (LSSD) rules, on three VLSI signal-processing chips. Our method (designated LSSD on-chip self-test, or LOCST) uses on-chip pseudorandom-pattern generation

Summary

A built-in self-test technique utilizing on-chip pseudorandom-pattern generation, on-chip signature analysis, a "boundary scan" feature, and an on-chip monitor test controller has been implemented on three VLSI chips by the IBM Federal Systems Division. This method (designated LSSD on-chip self-test, or LOCST) uses existing level-sensitive scan design strings to serially scan random test patterns to the chip's combinational logic and to collect test results. On-chip pseudorandom-pattern generation and signature analysis compression are provided via existing latches, which are configured into linear-feedback shift registers during the self-test operation. The LOCST technique is controlled through the on-chip monitor, IBM FSD's standard VLSI test interface/controller. Boundary scan latches are provided on all primary inputs and primary outputs to maximize self-test effectiveness and to facilitate chip I/O testing.

Stuck-fault simulation using statistical fault analysis was used to evaluate test coverage effectiveness. Total test coverage values of 81.5, 85.3, and 88.6 percent were achieved for the three chips with less than 5000 random-pattern sequences. Outstanding test coverage (>97%) was achieved for the interior logic of the chips. The advantages of this technique, namely very low hardware overhead cost (<2%), design-independent implementation, and effective static testing, make LOCST an attractive and powerful technique.

0740-7475/84/1100-0045$01.00 © 1984 IEEE

and on-chip signature analysis result compression. This is not a new self-test method; LOCST utilizes the serial-scan, random-pattern test technique pioneered by Eichelberger et al.[4,5] and Bardell et al.[6] of IBM. This article (1) details the adaptation of this technique to our existing chip testability architecture, (2) details the implementation of LOCST on three VLSI chips designed and fabricated by IBM FSD, and (3) discloses the results of the test coverage evaluations performed on these three chips. (For a thorough understanding of the principles of serial-scan, random-pattern testing, I strongly recommend a review of references 4, 5, and 6 and also a very comprehensive paper by Komonytsky.[7])

Standard FSD VLSI testability features

For a better understanding of the self-test architecture chosen for LOCST, a discussion of design features typical to IBM FSD's products is warranted. Figure 1 illustrates the three standard testability features incorporated in our VLSI products. They include

- level-sensitive scan design,
- "boundary scan" latches, and
- a standard maintenance interface, the on-chip monitor, or OCM.

All chips are designed following IBM's LSSD rules (see Eichelberger and Williams[8]) to ensure high test coverage and high diagnostic resolution during chip manufacture testing. "Boundary scan" is a requirement that all primary inputs (PIs) feed directly into shift register latches (SRLs, or LSSD latches) and all primary outputs (POs) are fed directly from SRLs. Boundary scan greatly simplifies chip-to chip interconnect testing and also provides an ideal buffer between LSSD VLSI products and non-LSSD vendor components, thereby reducing the complexity of testing "mixed-technology" cards.

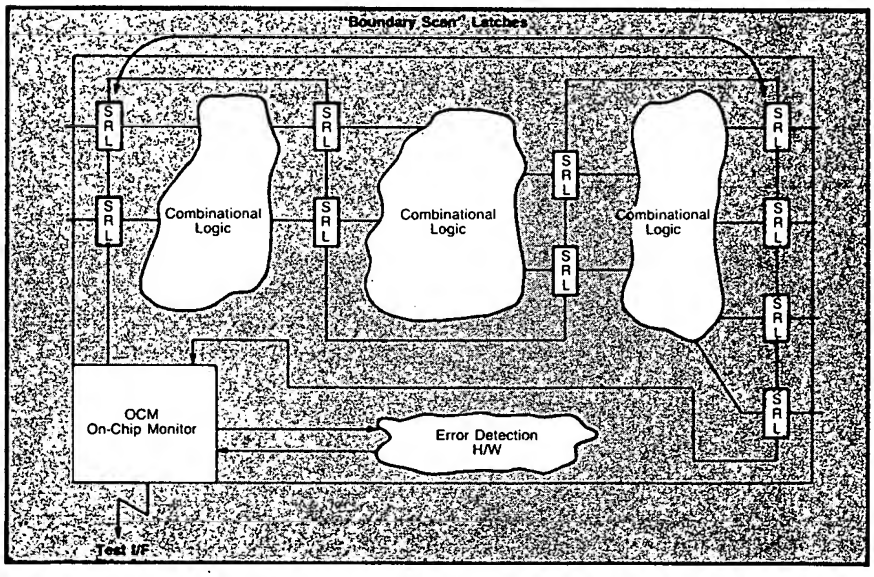

Figure 1. Standard VLSI features.

The OCM is a standard maintenance interface for our VLSI chips (Figure 2). It consists of seven lines: two for data transfer, four for control, and one for error reporting. The OCM maintenance bus can be configured as either a ring, a star, or a multidrop network, depending on system maintenance requirements. The four major functions of the OCM are

- scan string control,
- error monitoring and reporting,
- chip configuration control, and
- clock event control: run/stop, single cycle, and stop on error.

During LSSD testing (chip manufacture testing), scan strings are accessed via either dedicated or shared PIs and POs. (Note: The OCM is not used as a test aid during LSSD testing; it is simply logic to be tested by LSSD test patterns.) During card and system test, however, chip scan strings are accessed via the OCM interface.

The error detection hardware depicted in Figure 1 consists of on-chip error checkers used for on-line system error detection and/or fault isolation (described by Bossen and Hsiao[9]). When these checkers are triggered by

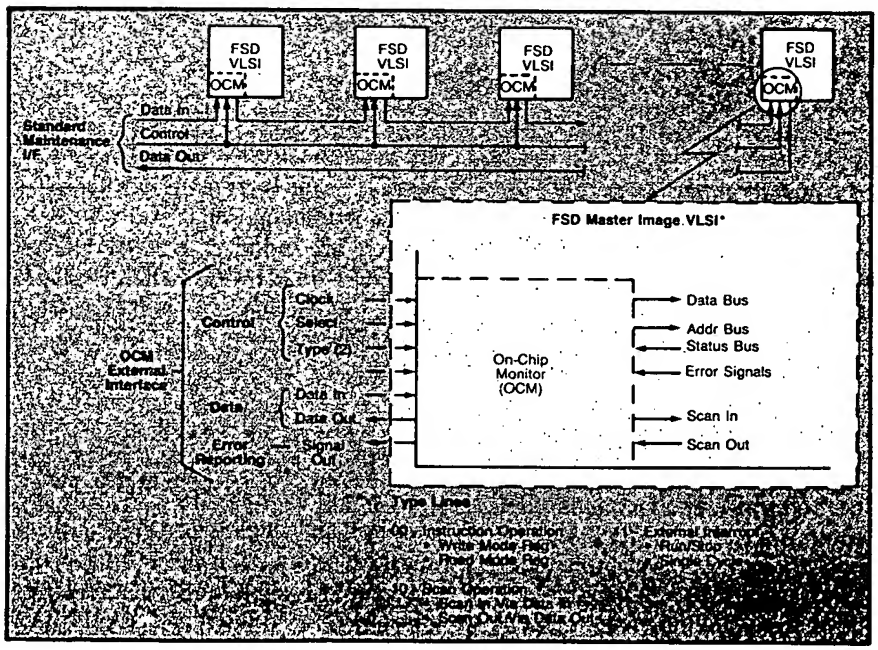

Figure 2. On-chip monitor.

an on-chip error, an attention signal is sent to the system maintenance processor through the OCM interface. The system maintenance processor reads internal chip error registers (or writes internal chip mode control registers) via OCM "instructions."

LOCST architecture

The basic self-test methodology used in LOCST is to (1) place pseudorandom data into all chip LSSD latches via serial scan, (2) activate system clocks for a single cycle to capture the results of the random-pattern stimuli through the chip's combinational logic, and (3) compress the captured test results into a pass-fail signature. With the existing testability features (LSSD, boundary scan, OCM) on each chip, it was a simple matter to incorporate a self-test capability.

To perform the pseudorandom-pattern-generation and signature-compression operations while in LOCST self-test mode, functional SRLs are reconfigured into linear-feedback shift registers, or LFSRs. The pseudorandom-pattern generator, or PRPG, is 20 bits in length, and the signature analyzer (SA) is 16 bits in length (see Figure 3). It should be noted that the devices shown in Figure 3 operate as normal serial-scan latches *and* as linear-feedback shift registers. The transformation from normal serial-scan mode to LFSR mode is controlled by multiplexing the scan inputs with a self-test enable signal (controlled via the OCM interface). The parallel data ports of these latches are not modified in any way. During self-test the data port clocks (system clocks) are disabled to prevent outside data from disrupting the deterministic sequences of the LFSRs.

The feedback polynomial for the PRPG was chosen because it is the least expensive "maximal-length" 20-bit LFSR implementation in terms of XOR gates required. For the LOCST implementation, the characteristic polynomial of the PRPG and the SA is fixed. Differing test pattern sequences can be obtained by altering the initial value (or "seed") of the PRPG. The feedback polynomial for the SA was chosen because of its proven performance (see Frohwerk[10] and Smith,[11] for example). The result of using a 20-bit PRPG and a 16-bit SA is a self-test capability with $2^{20} - 1$ possible random-pattern sequences and a very low probability of signature analysis fault masking (approximately $1/2^{16}$ or 0.0015 percent).

A high-level block diagram of the LOCST implementation structure is shown in Figure 4. In self-test mode the initial 20 SRLs of the chip's scan strings are configured into a PRPG LFSR, and the last 16 SRLs are configured into an SA LFSR. For normal LSSD chip manufacture testing, a chip usually contains several scan strings —each accessible from chip input and output pins. During LOCST testing, however, all scan strings except the one containing the OCM latches are configured into a single scan string. (Note: Random test patterns are scanned into the single scan string under OCM control. SRLs that are part of the OCM and any chip clock generation circuitry *cannot* be included in the LOCST scan string since self-test control and clock control cannot be disrupted by random data.)

The following is a description of the LOCST sequence:

(1) Initialize all internal latches: scan known data into all SRLs; this includes scanning "seeds" into PRPG and SA registers.

(2) Activate self-test mode: enable PRPG and SA registers; disables system clocks on input boundary SRLs and LFSRs.

(3) Perform self-test operation:

(a) Apply scan clocks until entire scan string (up to the SA LFSR) if filled with pseudorandom patterns. This step also scans test data into the SA LFSR for test result compression.

(b) Activate system clocks for single-cycle operation.

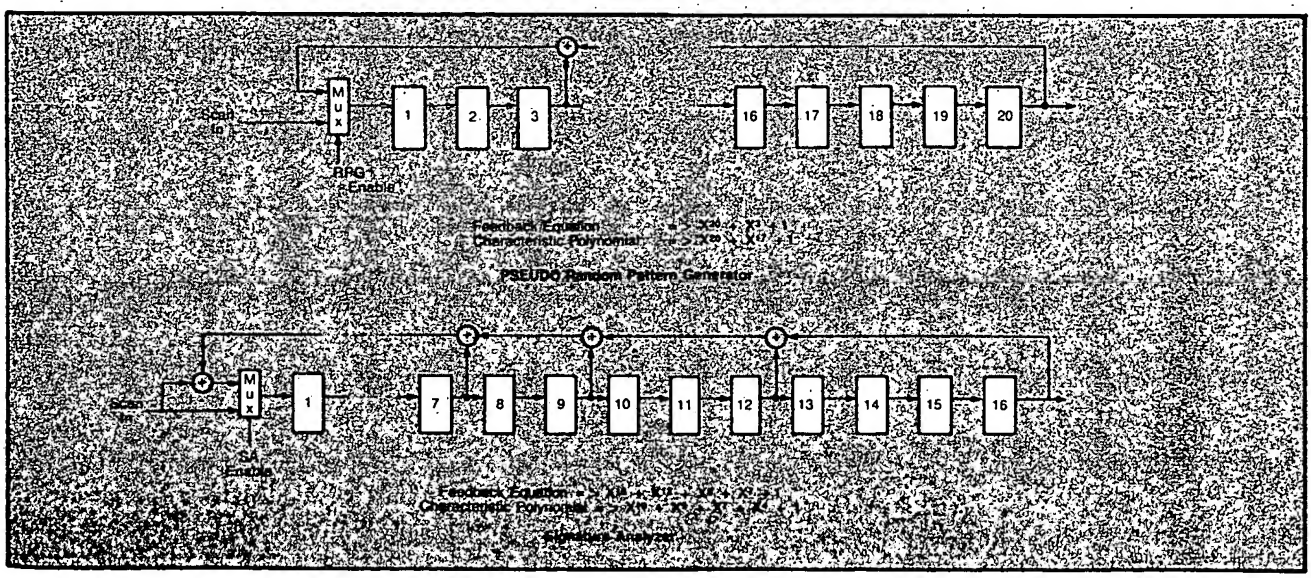

Figure 3. Linear-feedback shift register implementations.

(c) Repeat (a) and (b) until finished.

(4) Read out test result signature and compare with known "good" value.

The "good" value from step 4 can be obtained in two ways; (1) simulation of the entire self-test sequence, or (2) the "golden chip" approach (that is, determine what the "good" value is by performing the LOCST self-test operation on chips which have passed all other forms of manufacture and functional testing). Due to the high cost of the first method, the second is currently being used. If the correct "good" signature value were known (via simulation) during the chip design phase, a hardware comparator could be placed on the chip to provide an immediate pass-fail indication. Our implementations of LOCST require that the 16-bit signature be read by an external processor for comparison against the stored 16-bit "good" value.

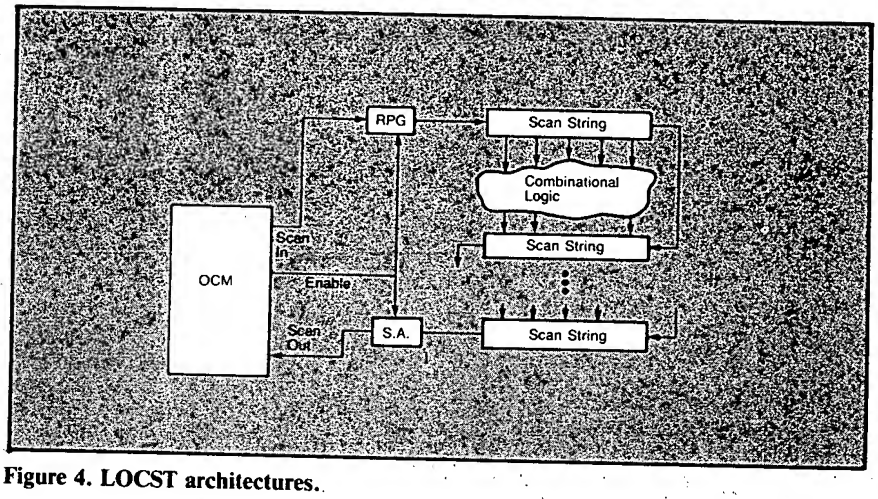

Figure 4. LOCST architectures.

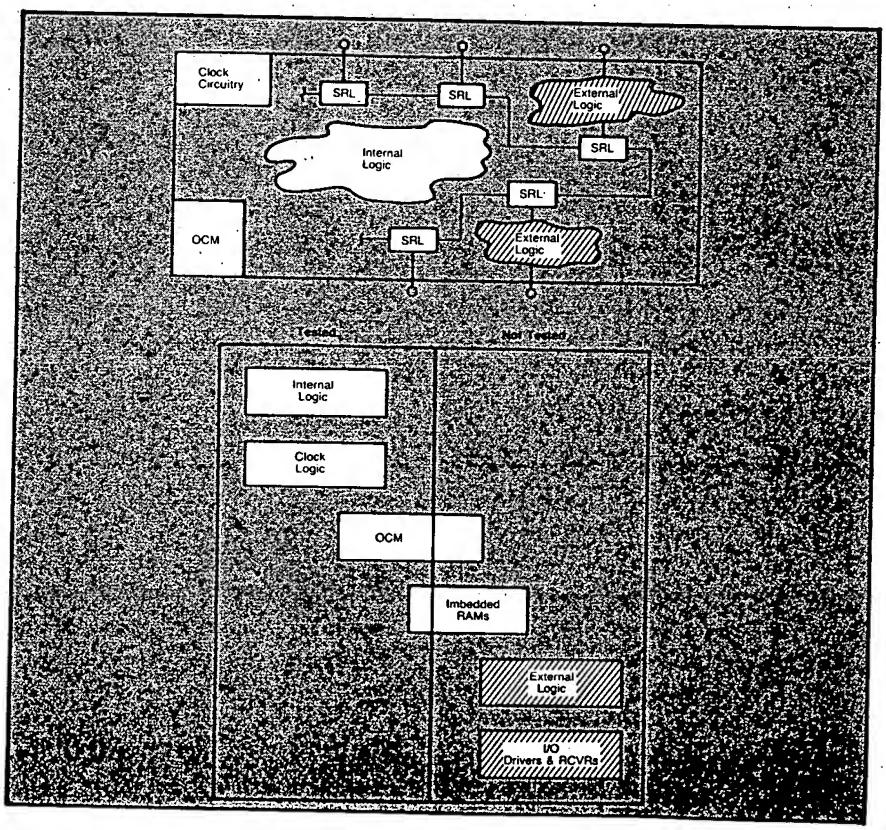

Figure 5. LOCST effectiveness.

The entire LOCST self-test operation is controlled by an external processor via the OCM interface. The external processor may be a chip or card tester or a system maintenance processor, depending on the testing environment. The OCM provides the following self-test control functions:

- PRPG and SA enable control,
- scan access to internal SRLs for random-pattern insertion and test result compression,
- chip clock control for single-cycle operation (if on-chip clock generation is used), and
- access to self-test results via direct register read or via scan.

If a chip does not have an OCM, control of these functions must be provided by some other means.

The data port clocks of input SRLs (i.e., boundary scan LSSD latches fed directly by primary inputs) are inhibited during self-test mode to prevent unknown data from corrupting the self-test sequence. If the input latch clocks are not disabled, known values *must* be ensured on chip PIs during self-test execution.

LOCST limitations

Like all on-chip self-test techniques, LOCST is incapable of testing the entire chip. In considering on-chip self-test effectiveness, we can divide chip logic into two basic categories: interior logic and exterior logic. Figure 5 illustrates the effectiveness of LOCST for the various chip regions. Since self-test patterns are applied via serial scan into chip latches, only the logic fed by latches will have random test patterns applied to it and test results will be captured only for logic which feeds latches. Chip logic whose inputs are fed by latches and whose outputs feed latches is designated "interior logic," and combinational logic whose inputs are fed by chip PIs and whose outputs feed chip POs is designated "exterior logic."

Obviously, external logic is completely untestable by the LOCST technique. The importance of boundary scan to on-chip self-test also should be obvious. The larger the percentage of exterior logic on a chip, the less effective on-chip self-test becomes. In the ideal case with 100 percent boundary scan, the only exterior logic would be I/O drivers and receivers (and with 100 percent boundary scan, I/O drivers and receivers would be very easy to test!)

Types of chip logic that do not clearly fall into the categories of interior or exterior are the OCM logic and embedded RAMs. Since the OCM controls the self-test operation, internal OCM logic is not tested by random patterns during self-test. Rather, the OCM is tested to the extent that all OCM functions needed to perform the self-test operation will have been exercised (i.e., scan control, clock control, loading self-test registers, etc.). Remaining OCM functions are tested by exercise of the OCM's remaining instruction set. RAMs embedded in a chip will not be completely tested by the LOCST self-test technique. Special RAM self-test circuitry would be needed to provide effective testing with random patterns. This topic is not addressed here.

The locations of the PPG and SA LFSRs are not illustrated because this would require a detailed scan string diagram. As mentioned previously, the PRPG and SA LFSRs utilize existing functional latches. The two other chips, B and C, when configured with a vendor multiply chip, perform digital filtering functions. Like Chip A, Chips B and C are primarily arithmetic data pipelines. All three chips are now incorporated in signal-processing systems.

To determine the testing effectiveness of the LOCST technique on these three chips, we performed fault simulation of the self-test procedure. Fault simulation provides a test coverage value upon which self-test effectiveness is based. The fault simulation was based on the classical stuck-fault model. Full fault simulation of the LOCST operation would have been too costly, so we followed this methodology:

- We used a statistical random sample of the full stuck-fault list. Test coverage results therefore have a 95 percent confidence level.
- Since no significant ($\ll 1\%$) error masking occurs due to the LFSR compression of the test results,[10-12] simulation of the serial compression activity of the SA LFSR was not performed. If the detection of a fault is observed at an SRL, it is assumed that this fault will be detected after LFSR compression.

We generated pseudorandom patterns placed in the latches during fault simulation via a PL/I program, using the same characteristic polynomial as the PRPG LFSR implemented on the chips (see Figure 3). A plot of test coverage vs. the number of self-test sequences for Chip A is presented in Figure 7. A total chip test coverage of 88.6 percent was achieved (with 95 percent confidence) with 3000 self-test sequences. Figure 8a displays the coverage evaluation results for Chip A in a different manner. Here Chip A's logic is divided into three categories (interior logic, exterior logic, and OCM logic) to highlight the LOCST testing effectiveness for each. LOCST test effectiveness for all three chips is summarized in Figure 8.

Implementions and coverage evaluation

The LOCST technique has been implemented on three VLSI chips used for signal-processing applications. The three chips—hereafter called Chip A,

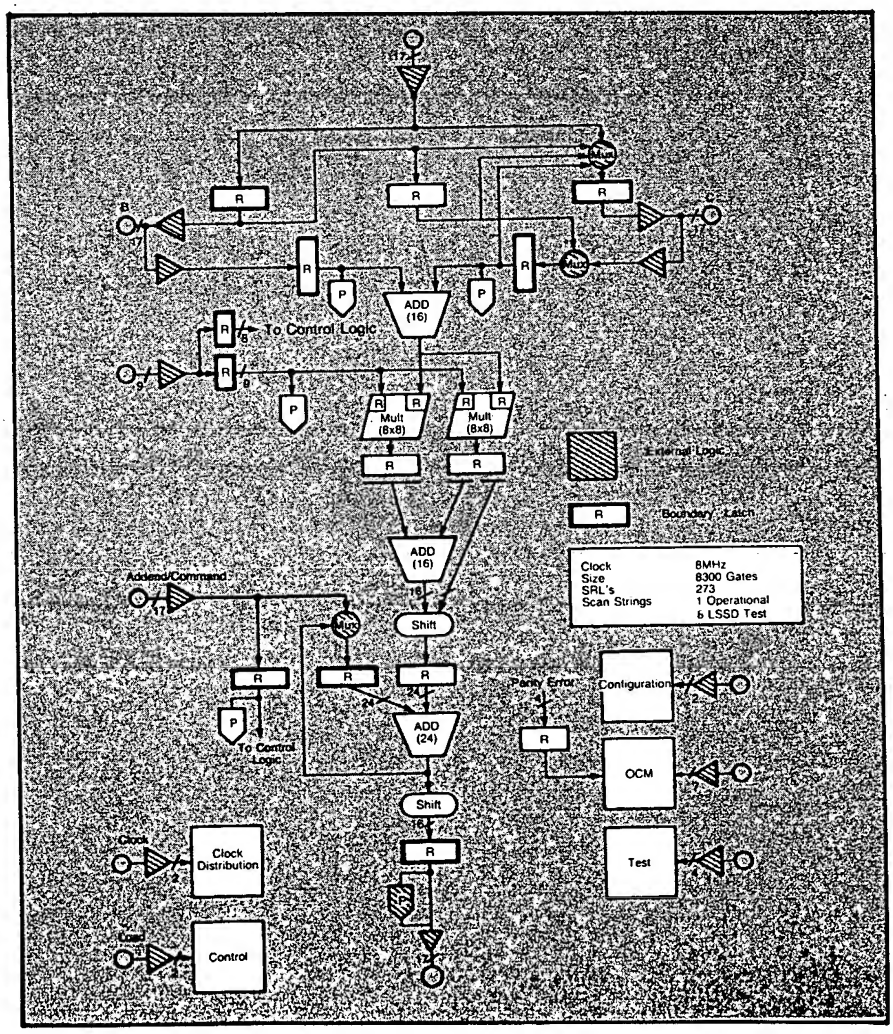

Figure 6. Signal-processing Chip A.

Chip B, and Chip C—were designed and fabricated in 1982. The addition of the LOCST capability (i.e., LFSRs for PRPG and SA functions and OCM self-test control logic) represents a hardware overhead of less than two percent. (Note: This figure does not include LSSD overhead or OCM overhead, as these features are included whether or not LOCST is implemented. Total testability overhead is in the 10-15 percent range.)

One of the three chips, Chip A, performs front-end signal-processing functions requiring high-rate, multiply-intensive algorithms such as finite-impulse response filtering, linear beam-forming, and complex band-shifting operations. Chip A performs these functions by utilizing a simple add-multiply-add pipelined data structure. A high-level diagram of Chip A is shown in Figure 6.

Overall test coverage values of 88.6, 81.5, and 85.3 percent (mean values of a 95 percent confidence interval) were obtained for the three chips respectively. Very good coverage (>97%) was obtained for the interior logic of all three chips with relatively few random-pattern loads (<5000). Test coverage obtained by deterministic LSSD test pattern generation was greater than 99 percent for all three chips. Whether or not test coverage comparable to that of LSSD testing could be obtained if more random-pattern loads were simulated (e.g., 10K, 100K, or 1M) was not evaluated because of the limited budget of this evaluation task.

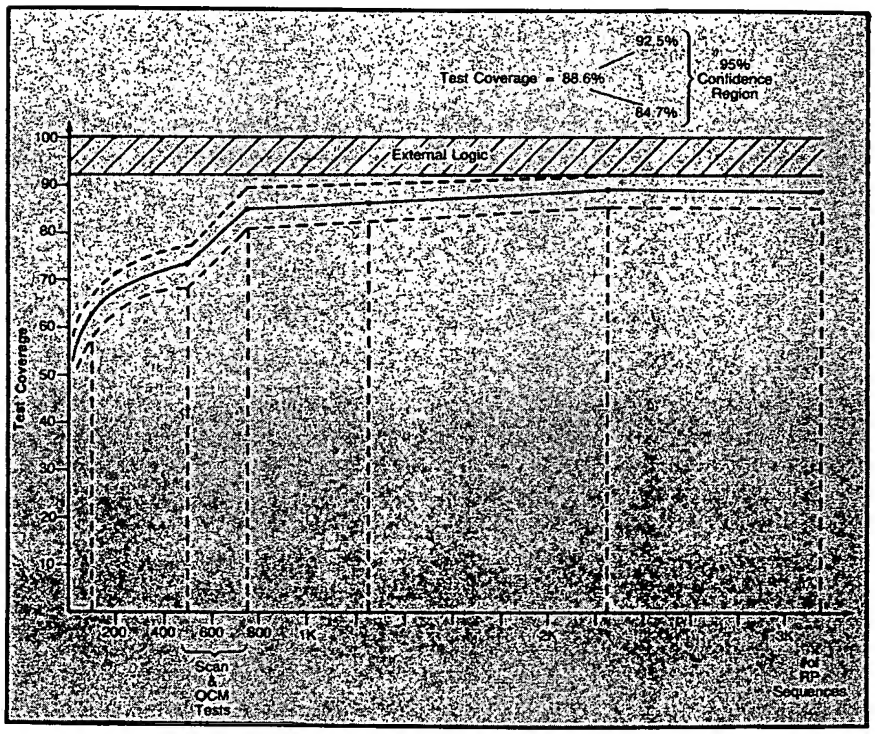

Figure 7. Test coverage results for Chip A.

Table 1.
LOCST test time.

	NO. OF RPs	NO. OF SRLs	TEST TIME
Chip A	2K	213	0.43s
Chip B	500	230	0.12s
Chip C	3K	223	0.67s

Test time = (No. of RPs/scan rate) × No. of SRLs
Scan rate = 1 MHz
RP = random-pattern sequence

LOCST execution time

In addition to providing high test coverage, a self-test technique should execute in a relatively short period of time. Table 1 presents the equation for calculating LOCST execution times and the predicted test times for the three FSD chips. For the assumed scan rate (based on existing FSD scan controllers) and the number of self-test sequences (based on the presented test coverage evaluation), subsecond execution times are achieved for all three chips.

If a large number of random-pattern loads is required to achieve adequate test coverage results, if the scan rate is slow (e.g., 1 MHz or less), or if a chip contains a large number of SRLs, LOCST self-test times may become quite large (minutes). An alternative to the basic LOCST implementation is to use many parallel scan strings feeding a multiple-input signature register, or MISR. This modification, illustrated in Figure 9, reduces the number of serial shifts required to fill all chip SRLs with random test data, thereby reducing the overall LOCST test time.

Self-test environments

One of the greatest potentials of self-test is the possibility of eliminating the need to produce a unique test pattern set for each test environment. The major test environments are

- chip manufacture test,
- card test,
- operational system test, and
- field return test (repair test).

The lack of defect diagnostic information is the key reason that self-test is not considered a viable technique for chip manufacture testing. But ongoing research is investigating the use of self-test techniques for LSI devices in the chip manufacture environment. A very promising technique using random-pattern testing for diagnosing failures has been developed by F. Mo-

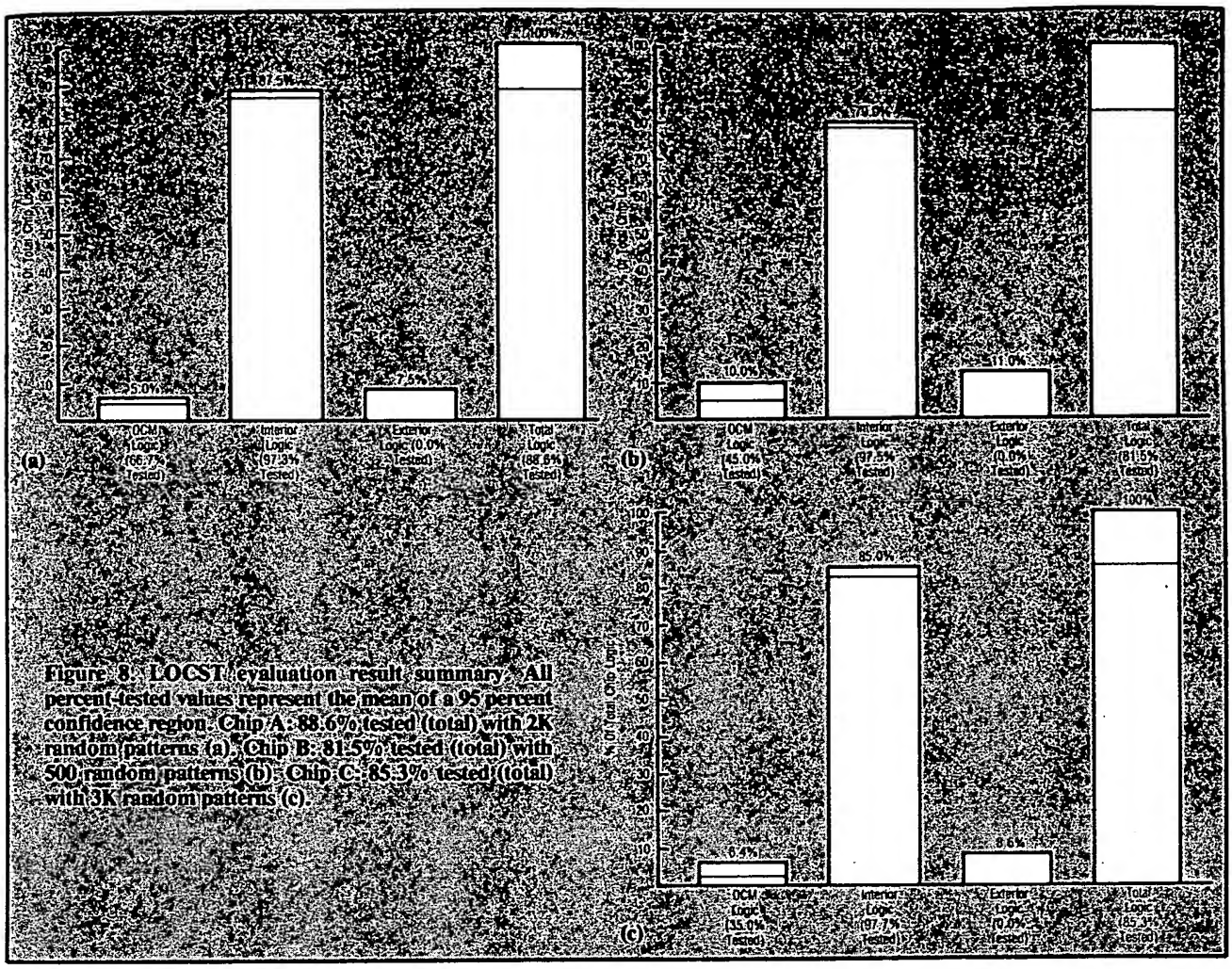

Figure 8. LOCST evaluation result summary. All percent-tested values represent the mean of a 95 percent confidence region. Chip A: 88.6% tested (total) with 2K random patterns (a). Chip B: 81.5% tested (total) with 500 random patterns (b). Chip C: 85.3% tested (total) with 3K random patterns (c).

tika et al.[13] of IBM Kingston. Presently, LOCST does not replace LSSD testing in the FSD chip manufacture test environment but is used as a supplemental chip-testing technique. As a minimum, since it provides a rapid pass-fail indication, self-testing would be useful in a production test environment to provide efficient preliminary screening of product.

The inclusion of several 10,000-gate VLSI components onto cards that have historically contained 5000 to 8000 gates of logic posed a serious problem to traditional card test methodologies. On-chip self-test offers a very effective solution. LOCST is used to verify that the FSD VLSI components on a card are defect-free. All FSD VLSI components are accessed via their OCM interface, requiring only seven card connector pins. Chip boundary scan latches (accessed via

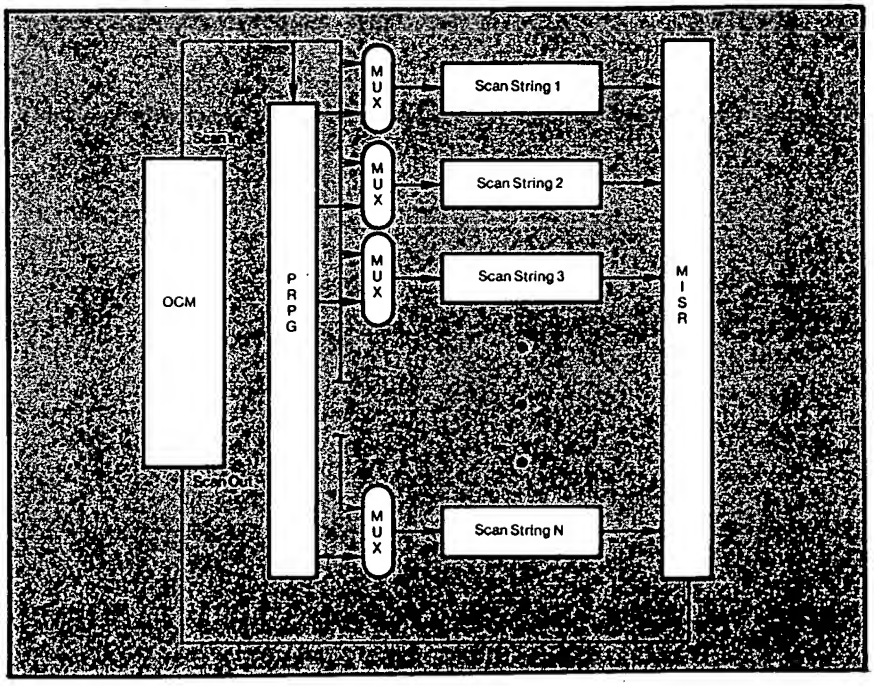

Figure 9. LOCST modification for faster execution.

the OCM) are used to apply and capture data for chip-to-chip interconnect testing. Boundary scan also effectively isolates FSD VLSI components from vendor components, enabling the use of traditional methods for testing the vendor logic on the card.

On-chip self-test supports the following types of operational system testing:

- system initialization test,
- system on-line periodic test, and
- system off-line fault localization test.

The objectives of implementing an on-chip self-test capability in our VLSI chips were to substantially reduce the plethora of unique test pattern sets for the differing test environments, reduce the volume of test vectors required to test our VLSI products, and eliminate the need for manual test pattern generation. Priorities of low hardware overhead, simple implementation, simple self-test control, high test coverage, and short self-test execution time were of prime importance. The implementation of LOCST on our VLSI products has enabled us to meet these objectives without violating any of the priorities. □

Acknowledgments

I would like to express my appreciation to E. B. Eichelberger and E. Lindbloom for their counsel and encouragement during the development and implementation of this self-test technique. I would also like to thank Tina Nguyen for her assistance with the fault simulation activity.

References

1. B. Koenemann, J. Mucha, and G. Zwiehoff, "Built-In Logic Block Observer," *Digest of Papers 1979 Test Conf. IEEE,* Oct. 1979, pp. 37-41.
2. R. M. Sedmak, "Design for Self-Verification: An Approach for Dealing with Testability Problems in VLSI-Based Designs," *Digest of Papers 1979 Test Conf. IEEE,* Oct. 1979, pp. 112-120.
3. E. J. McCluskey and S. Bozorgui-Nesbat, "Design for Autonomous Test," *IEEE Trans. Computers,* Vol. C-30, No. 11, Nov. 1981, pp. 866-875.
4. E. B. Eichelberger and E. Lindbloom, "Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test," *IBM J. Research and Development,* Vol. 27, No. 3, May 1983, pp. 265-272.
5. T. W. Williams and E. B. Eichelberger, "Random Patterns Within a Structured Sequential Logic Design," *Digest of Papers 1977 Semiconductor Test Symp. IEEE,* Oct. 1977, pp. 19-26.
6. P. Bardell and W. McAnney, "Self-Testing of Multichip Logic Modules," *Digest of Papers 1982 Int'l Test Conf. IEEE,* Nov. 1982, pp. 200-204.
7. D. Komonytsky, "LSI Self-Test Using Level Sensitive Scan Design and Signature Analysis," *Digest of Papers 1982 Int'l Test Conf. IEEE,* Nov. 1982, pp. 414-424.
8. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," *J. Design Automation and Fault Tolerant Computing,* Vol. 2, No. 2, May 1978, pp. 165-178.
9. D. C. Bossen and M. Y. Hsiao, "ED/FI: A Technique for Improving Computer System RAS," *Digest of Papers 11th Ann. Int'l Symp. Fault-Tolerant Computing,* June 1981, pp. 2-7.
10. R. A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," Hewlett-Packard Application Note 222-2, pp. 9-15.
11. J. E. Smith, "Measures of the Effectiveness of Fault Signature Analysis," *IEEE Trans. Computers,* Vol. C-29, No. 6, June 1980, pp. 510-514.
12. D. K. Bhavsar and R. W. Heckelman, "Self-Testing by Polynomial Division," *Digest of Papers 1981 Int'l Test Conf. IEEE,* Oct. 1981, pp. 208-216.
13. F. Motika, et al., "An LSSD Pseudo Random Pattern Test System," *Int'l Test Conf. 1983 Proc. IEEE.,* Oct. 1983, pp. 283-288.

Johnny J. LeBlanc is a staff engineer working in the System/Subsystem Testability department of the IBM Federal Systems Division in Manassas, Virginia, where he is responsible for the development of VLSI design for testability methodologies, including VLSI test generation, built-in test, and on-chip self-test.

LeBlanc received his BS in electrical engineering from the University of Southwestern Louisiana in 1976 and an MS in electrical engineering from North Carolina State University in 1978.

Questions concerning this article may be addressed to LeBlanc at Bldg. 400/044, IBM Federal Systems Division, 9500 Godwin Drive, Manassas, Va. 22110.